



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,889	09/30/2003	Shih-Wei Liao	42P16806	8089
8791 7590 09/11/2008 BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP 1279 OAKMEAD PARKWAY SUNNYVALE, CA 94085-4040				
EXAMINER MITCHELL, JASON D				
ART UNIT		PAPER NUMBER		
2193				
MAIL DATE		DELIVERY MODE		
09/11/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

**Application No.**

10/676,889

**Applicant(s)**

LIAO ET AL

**Examiner**

Jason Mitchell

**Art Unit**

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 14 January 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-29 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 9/30/03 and 1/14/08 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/S5108)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

This action is in response to an amendment filed on 1/14/08.

Claims 1-29 are pending in this case.

### ***Response to Arguments***

#### **Regarding the section labeled "Objections to the Drawings, Figures 1 and 2".**

Starting in 2<sup>nd</sup> par. the applicants state:

Office Action states "The figures show prior art systems usable with or by the claimed system, but do not show any structure representing an aspect of the claimed system" (Office Action, Page 2). It appears that the Office Action asserts the alleged admitted prior art without interpreting the drawings in view of the specifications. On the contrary, these drawings illustrate embodiments of the invention that together with the description serve to explain the principles of the invention (Specification, BRIEF DESCRIPTION OF THE DRAWINGS, [0005]).

Specifically, for example, Figure 1 contains an exemplary system performing the processes shown in Figures 5-8 (Specification, [0029]). A software architecture of a multithreading system executing one or more generated helper threads using parallelization analysis is included as illustrated in Figures 5-8 ([0011 ]-[0014]). Additionally, Figure 2 shows an embodiment of a computing system capable of performing disclosed techniques in the instant application, including, for example, speculative threads (Specification, [0038] [0039]). Apparently, the instant application discloses techniques including the claimed invention. Thus, Figures 1 and 2 are clearly part of the invention as claimed.

The examiner respectfully disagrees. While figs. 1 and 2 may be useful in explaining the process shown in Figures 5-8, figs. 1 and 2 do not show the process of figures 5-8. Specifically figure 1 shows a prior art computing system which can be used to perform the process shown in figures 5-8, but does not actually show the process being performed as applicant has asserted. (see e.g. applicants' par. [0029] "system 100 may be ... an Intel Pentium<sup>™</sup> 4 Hyper-Threading system"; par. [0030] "such details are not germane t the present invention.") Similarly figure 2 is disclosed as "a computing

system 200 capable of performing the disclosed techniques" (see par. [0038]) but does not in fact show the "disclosed techniques". Accordingly, figs. 1 and 2 are objected to as only illustrating that which is old. This objection can be overcome by pointing to an item or items illustrated in the drawings (i.e. Figs. 1-2) which the applicants assert are "new".

**Regarding the section labeled "Rejections under 35 U.S.C. § 102(b) Claims 1-2, 8-9 and 15-16".**

In the 3rd par. the applicants state:

Rather, Luk provides ... pre-execution by determining which references could benefit from the pre-execution and calculating the pre-execution distance (Luk, page 44, col. 2, par. 4). In addition, Luk's pre-execution thread stops either at a pre-determined PC (program counter) or when a sufficient number of instructions have been pre-executed (Luk, page 41, col. 2, par. 1). Thus, Luk determines to terminate pre-execution independent of where in the main thread the pre-execution is started. However, Luk is completely silent about identifying a region of a main thread to analyze the region for one or more helper threads

The examiner respectfully disagrees. First, the applicants have failed to point out how the "identifying a region of the main thread" language of the claims patentably distinguishes them from Luk's "determining which references could benefit from the pre-execution" and accordingly fail to comply with 37 CFR 1.111(b). Luk discloses a "locality analysis phase which determines [identifies] which references [regions] are likely to cause cache misses [have delinquent loads]" (pg. 44, col. 2 3<sup>rd</sup> full par.)

Second, the claims are not directed to termination of the helper threads. Accordingly the applicants' assertion that "Luk determines to terminate pre-execution independent of where in the main thread the pre-execution is started" does not

represent a patentable distinction. Specifically although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In the 5<sup>th</sup> par. the applicants state:

Specifically, Luk describes a local analysis which locates where cache misses occur in a program to determine if the misses are generated by access patterns that can potentially be pre-executed (Luk, page 5 l, col. 2, pars. Step1-Step2). Luk discusses using low-overhead profiling tools or cache simulation to locate cache misses (Luk, page 5 l, col. 2, par. Step 1). In addition, Luk uses compilers to recognize pointer-based data structures, array access, control-flow analysis and call-graph analysis (Luk, page 5 l, col. 2, par. Step 2). Apparently, Luk's locality analysis does not identify a region of a main thread that likely has one or more delinquent loads.

The examiner respectfully disagrees. Again, the applicants have failed to point out how the "identify[ing] a region" language of the claims patentably distinguishes them from Luk's "local analysis". Further, Luk's use of "low-overhead profiling tools", "cache simulation" and "compilers" to perform the "local analysis" does not represent a distinction over the claim which does not recite details of how the "identification" is achieved.

**Regarding the section labeled "Claims 17-19, 22-25 and 28-30".**

In the 2<sup>nd</sup> par. the applicants state:

Specifically, for example, independent claim 17, as amended, includes the limitations:

"executing a main thread of an application in a multi-threading system; and spawning more than one helper threads from the main thread to perform one or more computations for the main thread when the main thread enters a region having one or more delinquent loads, code of the one or more

helper thread being created separately from code of the main thread during a compilation of the main thread.  
(emphasis in original)

It is respectfully noted that as currently amended claim 17 (also claims 23 and 29) recite "spawning *one or more* helper threads" and not "*more than one*".

Starting in the 2<sup>nd</sup> par. the applicants state:

... Applicants' claim 17, as amended, includes the limitation of spawning one or more helper threads associated with codes generated separately from code of the main thread. It is respectfully submitted that Luk fails to disclose or suggest the noted limitations.

... Clearly, Luk's main thread and pre-execution threads are executed based on the same code. Nowhere does Luk teach or suggest spawning one or more helper threads associated with codes generated separately from code of the main thread.

Respectfully, these arguments have been considered but are moot in view of the new ground(s) of rejection. Specifically, Annavaram teaches creating separate code for the helper threads (see Abstract "generates the required dependence graphs at runtime ... executes these graphs to generate the data addresses"). Further, Luk explicitly suggests the combination (see Luk pg. 50, bridging the cols. "our approach and [Anavaram's] can be complementary").

### ***Drawings***

**Figures 1 and 2 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g).**

The figures show prior art systems usable with or by the claimed system, but do not show any structure representing an aspect of the claimed system (see. e.g. Fig. 3,

Compiler 308). Accordingly the drawings only show that with was known at the time of filing.

Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

**Claims 1-2, 8-9, 15-16 are rejected under 35 U.S.C. 102(b) as being anticipated by "Tolerating Memory Latency through Software-Controlled Pre-Execution in Simultaneous Multithreading Processors" by Luk (Luk).**

**Regarding Claim 1:** Luk discloses a method, comprising:

identifying a region of a main thread that likely has one or more delinquent loads, the one or more delinquent loads representing loads which likely suffer cache misses

during an execution of the main thread (pg. 44, col. 2 3<sup>rd</sup> full par. "locality analysis phase which determines which references are likely to cause cache misses"; also see Appendix Phase I, Step 1);

analyzing the region for one or more helper threads with respect to the main thread (pg. 44, col. 2 3<sup>rd</sup> full par. "locality analysis phase which determines which references ... could benefit from pre-execution"; also see the Appendix Phase I, Step 3); and

generating code for the one or more helper threads, the one or more helper threads being speculatively executed in parallel with the main thread to perform one or more tasks for the region of the main thread (pg. 44, col. 2, 3<sup>rd</sup> full par. "performs all necessary code transformations"; also see the Appendix Phase II).

**Regarding Claim 2:** The rejection of claim 1 is incorporated; further Luk discloses identifying the region comprises:

generating one or more profiles for cache misses of the region (pg. 43, the par. bridging the cols. "based on profiling information"; also see the Appendix, Phase I, Step 1 "This step can be accomplished through some low-overhead profiling tools"); and

analyzing the one or more profiles to identify one or more candidates for thread-based prefetch operations (pg. 43, the par. bridging the cols. "the compiler usually needs to heuristically decide how to prefetch ... based on profiling information").



**Regarding Claims 8-9:** Claims 8-9 recite a computer readable storage medium for instructing a computer to perform the methods of claims 1-2 and are addressed similarly.

**Regarding Claims 15-16:** Claims 15-16 recite a system for performing the method of claim 1 and are addressed similarly.

Additionally claim 16 recites and Luk discloses the process is executed by a compiler during a compilation of an application (pg. 44, col. 2, 3<sup>rd</sup> full par. "the compiler ... is responsible for inserting pre-execution").

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 3-4 and 10-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Tolerating Memory Latency through Software-Controlled Pre-Execution in Simultaneous Multithreading Processors" by Luk (Luk) in view of "Exploiting Hardware Performance Counters with Flow and Context Sensitive Profiling" by Ammons et al. (Ammons).**

**Regarding Claim 3:** The rejection of claim 2 is incorporated; further Luk discloses generating one or more profiles for an application (pg. 43, the par. bridging the cols. “decide how to prefetch ... based on profiling information”) but does not explicitly disclose executing the application with debug information or sampling cache misses and accumulating hardware counters for each static load.

Ammons teaches generating one or more profiles comprises:

executing an application associated with the main thread with debug information (pg. 86, col. 1, last full par. “a tool ... that instruments program executables”); and  
sampling cache misses and accumulating hardware counter for each static load of the region to generate the one or more profiles for each cache hierarchy (pg. 85 col. 2 1<sup>st</sup> full par. “exploits the hardware performance counters”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Luk’s profiling (pg. 43, the par. bridging the cols. “based on profiling information”) using Ammons methods (pg. 86, col. 1, last full par.; pg. 85 col. 2 1<sup>st</sup> full par.) Those of ordinary skill in the art would have been motivated to do so in order to achieve the improved profiling disclosed (Ammons pg. 85, col. 2, 1<sup>st</sup> full par. “extends profiling techniques in two new directions”).

**Regarding Claim 4:** The rejection of claim 3 is incorporated; further Luk discloses analyzing the one or more profiles comprises:

correlating the one or more profiles with respective source code based on the debug information (pg. 41, col. 1, the last partial par. "decide where to launch pre-execution in the program, based on ... cache miss profiling").

Luk does not disclose identifying top loads that contribute to cache misses.

Ammons teaches identifying top loads that contribute cache misses above a predetermined level as the delinquent loads (pg. 86, col. 2, 1<sup>st</sup> partial par. "1% of the paths ... account for 42 and 56% of the misses.").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to identify the top loads discussed in Ammons (pg. 86, col.2, 1<sup>st</sup> partial par.) in Luk's profile data (pg. 43, the par. bridging the cols. "based on profiling information"). Those of ordinary skill in the art would have been motivated to do so in order to balance the number of helper threads that are created with the effectiveness of each thread.

**Regarding Claims 10-11:** Claims 10-11 recite a computer readable medium for instructing a computer to perform the methods of claims 3-4 and are addressed similarly.

**Claims 5-7, 2-14, 17-19, 22-25 and 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Tolerating Memory Latency through Software-**

**Controlled Pre-Execution in Simultaneous Multithreading Processors” by Luk (Luk) in view of “Data Prefetching by Dependence Graph Precomputation” by Annavaram et al. (Annavaram).**

**Regarding Claim 5:** The rejection of claim 1 is incorporated; further Luk does not disclose building a dependent graph and performing slicing based on the dependent graph. Luk does disclose “a collection of schemes ... have been proposed to construct and pre-execute slices” (pg. 50, col. 1, the last partial par.)

Annavaram teaches building a dependent graph that captures data and control dependencies of the main thread (pg. 1 Abstract “efficiently generates the required dependence graphs”); and

performing a slicing operation on the main thread based on the dependent graph to generate the helper threads (pg. 1 Abstract “generate the data addresses of the marked load/store instructions”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate Annavaram’s dependent graph and associated slicing operation with Luk’s system. Those of ordinary skill in the art would have been motivated to do so because Luk discloses “[His] approach and [Annavaram’s] can be complementary” (see Luk pg. 50, col. 2, 1<sup>st</sup> partial par.)

**Regarding Claim 6:** The rejection of claim 5 is incorporated; further Luk discloses analyzing the region further comprises:

performing a scheduling between the main thread and the helper threads (pg. 44, col. 2, 3<sup>rd</sup> full par. "a scheduling phase"); and

determining a communication scheme between the main thread and the helper threads (pg. 44, Fig. 4 "Proposed instruction set extensions to support pre-execution").

**Regarding Claim 7:** The rejection of claim 6 is incorporated; further Luk discloses analyzing the region further comprises determining a synchronization period for the helper threads to synchronize the main thread and the helper threads, each of the helper threads performing its tasks within the synchronization period (pg. 46, col. 1, 2<sup>nd</sup> par. a pre-execution thread must be terminated if its next PC is out of the acceptable range").

**Regarding Claims 12-14:** Claims 12-14 recite a computer readable medium for instructing a computer to perform the methods of claims 5-7 and are addressed similarly.

**Regarding Claim 17:** Luk discloses a method, comprising:

executing a main thread of an application in a multi-threading system (pg. 40, col. 1, 2<sup>nd</sup> par. "single threads running on multithreaded processor"); and

spawning one or more helper threads from the main thread to perform one or more computations for the main thread when the main thread enters a region having one or more delinquent loads (pg. 40, col. 1, 2<sup>nd</sup> par. "spawning helper threads ... generates data addresses, on behalf of the main thread"), during a compilation of the main thread (pg. 44, col. 2, 3<sup>rd</sup> full par. "the compiler ... is responsible for inserting pre-execution ... performs all necessary code transformations").

Luk does not disclose code of the one or more helper threads begin created separately from code of the main thread.

Annavaram teaches one or more helper threads being created separately from code of the main thread (Abstract "generates the required dependence graphs at runtime ... executes these graphs to generate the data addresses").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate Annavaram's dependent graph and associated slicing operation with Luk's system. Those of ordinary skill in the art would have been motivated to do so because Luk discloses "researchers have investigated ways to pre-execute only a subset of instructions ... our approach and [Annavaram's] can be complementary" (see Luk pg. 50, bridging the cols.)

**Regarding Claim 18:** The rejection of claim 17 is incorporated; further Luk discloses:

creating a thread pool to maintain a list of thread contexts (pg. 42, col. 1, 1<sup>st</sup> full par. "N hardware contexts supported by the machine"); and

allocating one or more thread contexts from the thread pool to generate the one or more helper threads (pg. 41, col. 1, the last partial par. "Each thread-spawning instruction requests for an idle hardware context to pre-execute the code sequence").

**Regarding Claim 19:** The rejection of claim 18 is incorporated; further Luk discloses:

terminating the one or more helper threads when the main thread exits the region (pg. 46, col. 1, 3<sup>rd</sup> par. "terminate a pre-execution thread if ... the main thread has executed N instructions after passing P"); and

releasing the thread contexts associated with the one or more helper threads back to the thread pool (pg. 41, col. 2, the 1<sup>st</sup> partial par. "T will free its hardware context").

**Regarding Claim 22:** The rejection of claim 17 is incorporated; further Luk discloses discarding results generated by the one or more helper threads when the main thread exits the region, the results not being reused by another region of the main thread (pg. 41, col. 2, the 1<sup>st</sup> partial par. "results held in T's registers are simply discarded").

**Regarding Claims 23-25 and 28:** Claims 23-25 and 28 recite a computer readable storage medium for instructing a computer to perform the methods of claims 17-19 and 22 and are addressed similarly.

**Regarding Claim 29:** Claim 29 recites a system for performing the method of claim 17 and is addressed similarly.

**Claims 20-21 and 26-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Tolerating Memory Latency through Software-Controlled Pre-Execution in Simultaneous Multithreading Processors” by Luk (Luk) in view of “Data Prefetching by Dependence Graph Precomputation” by Annavaram et al. (Annavaram) in view of US 7,243,267 to Klemm et al. (Klemm).**

**Regarding Claim 20:** The rejection of claim 17 is incorporated; further Luk discloses determining a period for each of the helper threads, each of the helper threads being terminated when the respective period expires (pg. 46, col. 1, 2<sup>nd</sup> par. “Once this limit is reached, the thread will be terminated anyway”).

The Luk-Annavaram combination does not explicitly disclose the period is a time period.

Klemm teaches determining a time period for a thread (col. 5, lines 57-58 “thread execution time exceeds user-specified threshold”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to terminate one of Luk’s helper threads after a time period expires (Klemm



col. 5, lines 57-58 "thread execution time exceeds user-specified threshold") as an alternate or additional instance of Luk's "system-enforced terminating conditions for preserving correctness or avoiding wasteful computation" (col. 46, col. 1, 1<sup>st</sup> par.)

**Regarding Claim 21:** The rejection of claim 20 is incorporated; further Luk discloses each of the helper threads terminates when the period expires even if the respective helper thread has not been accessed by the main thread (pg. 46, col. 1, 2<sup>nd</sup> par. "the thread will be terminated anyway").

**Regarding Claims 26-27:** Claims 26-27 recite a computer readable medium for instructing a computer to perform the methods of claims 20-21 and are addressed similarly.

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/  
Jason Mitchell

/Lewis A. Bullock, Jr./  
Supervisory Patent Examiner, Art Unit 2193